



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/720,960	11/24/2003	Richard D. Dettinger	ROC920030277US1	5201
46797 7590 11/14/2007 IBM CORPORATION, INTELLECTUAL PROPERTY LAW DEPT 917, BLDG. 006-1 3605 HIGHWAY 52 NORTH ROCHESTER, MN 55901-7829			EXAMINER DWIVEDI, MAHESH H	
			ART UNIT 2168	PAPER NUMBER
			MAIL DATE 11/14/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

MAILED

NOV 13 2007

Technology Center 2100

Application Number: 10/720,960
Filing Date: November 24, 2003
Appellant(s): DETTINGER ET AL.

Randol W. Read
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 08/03/2007 appealing from the Office action mailed 11/01/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,453,353	Win et al.	09/17/2002
7,027,975	Pazandak et al.	04/11/2006

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

2. Claims 10-12, 14-21, and 27-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Win et al.** (U.S. Patent 6,453,353) and in view of **Pazandak et al.** (U.S. Patent 7,027,975).

3. Regarding claim 10, **Win** teaches a method comprising:

- A) assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database (Win, Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
- B) wherein the assigned metadata requirements specify conditions required for successful execution of the functional module (Win, Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);

- C) obtaining a list of functional modules that are accessible from within the application used during the query session (Win, Abstract, Column 6, lines 10-16, lines 41-65);
- D) identifying a limited subset of the functional modules in the list that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements (Win, Abstract, Column 6, lines 10-16, lines 41-65); and
- E) providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute (Win, Abstract, Column 6, lines 10-16, lines 41-65).

The examiner notes that **Win** teaches **“assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database”** as, “Roles determine what resources a User can access. Further, each role may require a set of information that is available in resources” (Column 5, lines 44-46) and “When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that the user is authorized to access the resource” (Column 6, lines 58-64). The examiner further notes that **Win** teaches **“wherein the assigned metadata requirements specify conditions required for successful execution of the functional module”** as “the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource” (Abstract), “If the name and password are correct, the Authentication Client Module reads the user’s roles from the Registry server” (Column 6, lines 44-46), and “a personalized menu is an HTML page containing a list of authorized Resources” (Column 6, lines 13-14). The examiner further notes that **Win** teaches **“obtaining a list of functional modules that are accessible from within the application used during the query session”** as “When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that the user is authorized to access the resource” (Column 6, lines 58-64). The examiner further notes that **Win**

Art Unit: 2168

teaches **“identifying a limited subset of the functional modules in the list that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements”** as “a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access” (Column 6, lines 12-15) and “When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that the user is authorized to access the resource” (Column 6, lines 58-64). The examiner further notes that Win teaches **“providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute”** as “a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access” (Column 6, lines 12-15).

Win does not explicitly teach:

- F) collecting runtime metadata relating to the query session; and
- G) wherein the metadata is collected after the composition of a query.

Pazandak, however, teaches **“collecting runtime metadata relating to the query session collecting runtime metadata relating to the query session”** as “LL interface descriptor (B), as defined in more detail below, which is sent to the Parser Farm 308 on the server below” (Column 12, lines 19-22) and “The Parser 310 can also send the set or a subset of the Interface Descriptor 306, e.g., LL Parser ID, transaction ID, or other metadata” (Column 17, lines 16-20) and **“wherein the metadata is collected after the composition of a query”** as “LL interface descriptor (B), as defined in more detail below, which is sent to the Parser Farm 308 on the server below” (Column 12, lines 19-22) and “The Parser 310 can also send the set or a subset of the Interface Descriptor 306, e.g., LL Parser ID, transaction ID, or other metadata” (Column 17, lines 16-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching

Pazandak's would have allowed **Win's** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 11, **Win** does not explicitly teach a method comprising:
A) wherein the runtime metadata comprises attributes of fields involved in a query or query results.

Pazandak, however, teaches "**wherein the runtime metadata comprises attributes of fields involved in a query or query results**" as "Sentence Construction" (Column 14, line 4, Figures 7-9, 11), and "Choice items " (Column 15, line 2, Figures 7-9, 11).

The examiner notes that the various screenshots depicted in Figures 7-9, and 11 all depict several attributes associated with various fields of query construction.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak's** would have allowed **Win's** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 12, **Win** does not explicitly teach a method comprising:
A) wherein the runtime metadata comprises content contained in query results.

Pazandak, however, teaches "**wherein the runtime metadata comprises content contained in query results**" as "execution results of a sample LL query in a tabular format" (Column 32, lines 48-49, Figure 18).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak's** would have allowed **Win's** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 14, **Win** further teaches a method comprising:

A) wherein obtaining metadata associated with the functional module comprises examining a signature validation (Column 6, lines 1-3, Column 14, lines 34-43).

The examiner notes that **Win** teaches **“wherein obtaining metadata associated with the functional module comprises examining a signature validation”** as “users may log in either with a digital certificate or by opening a login page URL with a web browser and entering a name and password” (Column 6, lines 1-3).

Regarding claim 15, **Win** further teaches a method comprising:

A) wherein the metadata associated with at least one of the functional modules comprises at least one of: one or more input parameters required for successful execution of the functional module, one or more output parameters required for successful execution of the functional module, and a security credential required to execute the functional module (Abstract, Column 6, lines 10-16, lines 41-65).

The examiner notes that **Win** teaches **“wherein the metadata associated with at least one of the functional modules comprises at least one of one or more input parameters required for successful execution of the functional module; one or more output parameters required for successful execution of the functional module; and a credential of a user authorized to execute the functional module”** as “The Authentication Client Module and Access Menu Module authenticates a user by verifying the name and password with the Registry Server 108” (Column 6, lines 42-44).

Regarding claim 16, **Win** does not explicitly teach a method comprising:

A) wherein at least one of the functional modules analyzes query results.

Pazandak, however, teaches **“wherein at least one of the functional modules analyzes query results”** as “a sophisticated extension is a capability for cooperative response, where if the user’s query results in a request for which no items are returned (it is overly constrained), then the translation tree can be examined to examine the

results” (Column 32, lines 65-67-Column 33, line 1), and “A further sophisticated extension is to automatically relax some criteria” (Column 33, lines 9-10)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak’s** would have allowed **Win’s** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 17, **Win** does not explicitly teach a method comprising:

- A) the runtime metadata comprises the names of fields in a result set; and
- B) the limited subset of functional modules comprises functional modules requiring data from fields in the result set as inputs.

Pazandak, however, teaches “**the runtime metadata comprises the names of fields in a result set**” as “Categorization-Group items into (possibly hierarchical) sets of categories, whereby the user selects a category and is then shown the list of subcategories or choice items” (Column 14, lines 64-67, Figures 7-9, 11), and “**the limited subset of functional modules comprises functional modules requiring data from fields in the result set as inputs**” as “Categorization-Group items into (possibly hierarchical) sets of categories, whereby the user selects a category and is then shown the list of subcategories or choice items” (Column 14, lines 64-67, Figures 7-9, 11).

The examiner notes that the subcategories depicted in Figure 8 are analogous to selecting various fields to further limit a query.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak’s** would have allowed **Win’s** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 18, **Win** does not explicitly teach a method comprising:

Art Unit: 2168

A) wherein the runtime metadata comprises information related to a query building session.

Pazandak, however, teaches “**wherein the runtime metadata comprises information related to a query building session**” as “Appearance during sentence construction” (Column 14, line 4, Figures 1-9, 11).

The examiner notes that the various interfaces depicted in Figures 7-9, and 11 show query construction.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak’s** would have allowed **Win’s** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 19, **Win** does not explicitly teach a method comprising:

- A) wherein the information related to the query building session comprises a specified focus of the query; and
- B) identifying a limited subset of the functional modules that will successfully execute comprises identifying functional modules associated with the specified focus.

Pazandak, however, teaches “**wherein the information related to the query building session comprises a specified focus of the query**” as “Categorization-Group items into (possibly hierarchical) sets of categories, whereby the user selects a category and is then shown the list of subcategories or choice items” (Column 14, lines 64-67, Figures 7-9, 11), and “**identifying a limited subset of the functional modules that will successfully execute comprises identifying functional modules associated with the specified focus**” as “Categorization-Group items into (possibly hierarchical) sets of categories, whereby the user selects a category and is then shown the list of subcategories or choice items” (Column 14, lines 64-67, Figures 7-9, 11).

The examiner notes that it is common knowledge that in a hierarchical category organization, selected different categories yields different subcategories. The examiner further notes that a category is analogous to a “**focus**”.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak's** would have allowed **Win's** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 20, **Win** teaches a method and computer readable medium comprising:

- A) assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database (Win, Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
- B) wherein the assigned metadata requirements specify conditions required for successful execution of the functional module (Win, Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
- C) obtaining a list of functional modules accessible from within the application (Win, Abstract, Column 6, lines 10-16, lines 41-65);
- D) identifying a limited subset of the functional modules that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements (Win, Abstract, Column 6, lines 10-16, lines 41-65); and
- E) providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute (Win, Abstract, Column 6, lines 10-16, lines 41-65).

The examiner notes that **Win** teaches **"assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database"** as "Roles determine what resources a User can access. Further, each role may require a set of information that is available in resources" (Column 5, lines 44-46) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to

Art Unit: 2168

access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches **"wherein the assigned metadata requirements specify conditions required for successful execution of the functional module"** as "the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource" (Abstract), "If the name and password are correct, the Authentication Client Module reads the user's roles from the Registry server" (Column 6, lines 44-46), and "a personalized menu is an HTML page containing a list of authorized Resources" (Column 6, lines 13-14). The examiner further notes that **Win** teaches **"obtaining a list of functional modules accessible from within the application"** as "a list of authorized resources" (Column 6, lines 13-14) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches **"identifying a limited subset of the functional modules that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements"** as "a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches **"providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute"** as "a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15).

Win does not explicitly teach:

F) collecting runtime metadata relating to a query session.

Pazandak, however, teaches “collecting runtime metadata relating to a query session” as “LL interface descriptor (B), as defined in more detail below, which is sent to the Parser Farm 308 on the server below” (Column 12, lines 19-22) and “The Parser 310 can also send the set or a subset of the Interface Descriptor 306, e.g., LL Parser ID, transaction ID, or other metadata” (Column 17, lines 16-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak’s** would have allowed **Win’s** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 21, **Win** does not explicitly teach a computer readable medium comprising:

A) wherein the application is a query building application.

Pazandak, however, teaches “wherein the application is a query building application” as “Appearance during sentence construction” (Column 14, line 4, Figures 1-9, 11).

The examiner notes that the various interfaces depicted in Figures 7-9, and 11 show query construction.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak’s** would have allowed **Win’s** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 27, **Win** teaches a data processing system comprising:

A) a data repository (Win, Column 5, lines 13-15);

B) a plurality of functional modules, each having associated metadata requirements that specify conditions required for successful execution of the functional modules (Win, Abstract, Column 6, lines 10-16, lines 41-65);

Art Unit: 2168

C) an application from which the functional modules are accessible (Win, Abstract, Column 6, lines 10-16, lines 41-65);

D) present to a user a limited subset of the functional modules that will successfully execute, as determined by the application based on the collected runtime metadata and the metadata requirements associated with the functional modules (Win, Abstract, Column 5, lines 66-67-Column 6, lines 1-16).

The examiner notes that Win teaches **“a data repository”** as “The system 2 enables organizations to register information sources or Resources and register Users of the information in a central repository” (Column 5, lines 13-15). The examiner further notes that Win teaches **“a plurality of functional modules, each having associated metadata requirements that specify conditions required for successful execution of the functional modules”** as “a list of authorized resources” (Column 6, lines 13-14). The examiner further notes that Win teaches **“an application from which the functional modules are accessible”** as “a personalized menu is an HTML page containing a list of authorized Resources” (Column 6, lines 13-14). The examiner further notes that Win teaches **“present to a user a limited subset of the functional modules that will successfully execute, as determined by the application based on the collected runtime metadata and the metadata requirements associated with the functional modules”** as “the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource” (Abstract), “If the name and password are correct, the Authentication Client Module reads the user’s roles from the Registry server” (Column 6, lines 44-46), and “a personalized menu is an HTML page containing a list of authorized Resources” (Column 6, lines 13-14).

Win does not explicitly teach:

E) wherein the application is configured to collect runtime metadata after the composition of a query.

Pazandak, however, teaches **“wherein the application is configured to collect runtime metadata after the composition of a query”** as “LL interface descriptor (B), as defined in more detail below, which is sent to the Parser Farm 308 on

the server below" (Column 12, lines 19-22) and "The Parser 310 can also send the set or a subset of the Interface Descriptor 306, e.g., LL Parser ID, transaction ID, or other metadata" (Column 17, lines 16-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak's** would have allowed **Win's** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 28, **Win** does not explicitly teach a data processing system comprising:

A) wherein the data repository comprises XML data structures used to store runtime metadata.

Pazandak, however, teaches "**wherein the data repository comprises XML data structures used to store runtime metadata**" as "The LL Client code can take any of several forms: a scripted document using scripting language such as Javascript and/or dynamic HTML an otherwise encoded document using a format such as XML, or an application-specific encoding an HTML document" (Column 16, line 54-64) and "Properties can be encoded in various ways; for instance in XML" (Column 24, lines 62-63).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak's** would have allowed **Win's** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 29, **Win** further teaches a data processing system comprising:
A) wherein the data repository comprises relational database tables used to store runtime metadata (Column 5, lines 13-15, Column 7, lines 1-6).

The examiner notes that Win teaches **“wherein the data repository comprises relational database tables used to store runtime metadata”** as “The Registry Repository is structured as a database. For example, the Registry Repository may be an SQL Server relational database management system, the Oracle 7® database, etc.” (Column 7, lines 1-6). The examiner further notes that it is common knowledge that relational databases store data in tables.

(10) Response to Argument

A. Claims 10-12, 14-21, and 27-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Win et al. (U.S. Patent 6,453,353) and in view of Pazandak et al. (U.S. Patent 7,027,975).

1. Independent Claims 10, 20, and 27:

Arguments (1):

I) Regarding Independent Claims 10, 20, and 27, Appellant argues that the “Win does not disclose assigned metadata requirements that specify conditions for successful execution of a functional module”.

However, the instant application defines metadata as “Metadata based on managed data includes descriptive information about the context and characteristics of data managed within the system. Security oriented user information, such as user credentials, and user invoked session information also constitutes metadata” (Paragraph 26). Furthermore, the instant application defines the execution of functional modules as “As illustrated, when the query session is initiated, runtime metadata 170 may include session information 172, such as the user's login ID, the user's security credentials and other information obtained while a user's session remains active. Upon

session initiation, a set of plug-ins may be presented to the user to begin performing functions. The initial set of plug-ins may be limited to those where the plug-ins' requirements are met by the runtime metadata 170. For example, if the session information 172 within the runtime metadata 170 shows that a user is authorized to access a certain plug-in, that plug-in will be among the limited subset of plug-ins made available to the user upon session initiation" (Paragraph 48). Moreover, Columns 6 and 18 of **Win** describe the successful execution of resources (i.e. functional modules) as "a personalized menu is an HTML page containing a list of authorized Resources. In one embodiment, a Personalized Menu is an HTML page containing a list of authorized Resources. The Personalized Menu displays only Resources to which the User has access. The User can then select and access a Resource" (Column 6, lines 13-17), "If the name and password are correct, the Authentication Client Module reads the user's roles from the Registry server" (Column 6, lines 44-46), and "By these actions, the user specifies that persons having the role of "Sales Manager" are authorized to view or use the "National Sales Report" resource. Any user who is assigned the role of "Sales Manager" in the future will automatically have access to the "National Sales Report" resource. If the administrator later un-assigns "Sales Manager" from the "National Sales Report" resource, then all users associated with the "Sales Manager" role will automatically lose access to the resource" (Column 18, lines 28-36). The examiner wishes to state that the instant application states that security credentials (i.e. metadata) of a current session are used by the system to only execute the appropriate functional modules based on those security credentials (see "runtime metadata 170 shows that a

user is authorized to access a certain plug-in, that plug-in will be among the limited subset of plug-ins made available to the user upon session initiation"). The examiner further wishes to state that **Win's** method also only executes a resource (i.e. functional module) when the appropriate metadata requirements (user name/password and user role (i.e. security credentials)) are met (see "The Personalized Menu displays only Resources to which the User has access"). Therefore, the user roles of Win teach executing a functional module based on metadata.

II) Regarding Independent Claims 10, 20, and 27, Appellant argues that the "a role does not specify conditions for the execution of a particular resource" and "successful execution of a functional module is not the same as limiting a set of available resources based on a user's role...metadata requirements recited in the claims specify conditions for the successful execution of a functional module. See the last three sentences of Paragraph 44".

However, according to the last three sentences of Paragraph 44 ("In order for a plug-in to be successfully called, or executed, its requirements described by associated metadata 160 needs to be met by parameters available to the calling application. If the available parameters match all the plug-in's associated metadata 160, that plug-in will be placed on the list of available plug-ins 129. Further, some plug-ins may only be available to certain authorized users determined by session information 172 available in the runtime metadata 170"), a plug-in will only be called (i.e. executed) if to only authorized users based on metadata (i.e. security credentials). The examiner wishes to refer to Columns 6 and 18 of **Win** describe the successful execution of resources (i.e.

functional modules) as “a personalized menu is an HTML page containing a list of authorized Resources. In one embodiment, a Personalized Menu is an HTML page containing a list of authorized Resources. The Personalized Menu displays only Resources to which the User has access. The User can then select and access a Resource” (Column 6, lines 13-17), “If the name and password are correct, the Authentication Client Module reads the user’s roles from the Registry server” (Column 6, lines 44-46), and “By these actions, the user specifies that persons having the role of “Sales Manager” are authorized to view or use the “National Sales Report” resource. Any user who is assigned the role of “Sales Manager” in the future will automatically have access to the “National Sales Report” resource. If the administrator later un-assigns “Sales Manager” from the “National Sales Report” resource, then all users associated with the “Sales Manager” role will automatically lose access to the resource” (Column 18, lines 28-36). The examiner wishes to state that the instant application states that security credentials (i.e. metadata) of a current session are used by the system to only execute the appropriate functional modules based on those security credentials (see “runtime metadata 170 shows that a user is authorized to access a certain plug-in, that plug-in will be among the limited subset of plug-ins made available to the user upon session initiation”). Because the instant application describes the calling (i.e. executing) of a functional module based on the security credentials of a specific user, the user roles which determine which resources will be called in **Win** teaches the limitation of assigned metadata requirements that specify conditions for successful execution of a functional module.

Furthermore, Figure 4a of the instant application depicts functional modules that are called (i.e. executed) based on a user's role (see list 430 that corresponds to the nurse). Figure 10B of **Win** depicts that the user "sales manager" may access the resource (i.e. functional module) of the national sales report. Because both Figure 4a of the instant application and Figure 10B of **Win** depict calling functional modules only to authorized users, **Win's** method of presenting functional modules to authorized users based on metadata teaches the assigned of metadata requirements that specify conditions for successful execution of a functional module.

Arguments (2):

A) Regarding Independent Claims 10, 20, and 27, Appellant argues that the "the combination of Win and Pazandak does not disclose collecting runtime metadata relating to a query session after the composition of the query". Specifically, Appellant argues that "However, to rely on Pazandak, runtime metadata must be interpreted as "End user selection from menu choices," and not as "user roles"...Were the two references combined, the collected runtime metadata could not be both "user roles" and "End-user selection from menu choices".

However, the instant application defines metadata as "Metadata based on managed data includes descriptive information about the context and characteristics of data managed within the system. Security oriented user information, such as user credentials, and user invoked session information also constitutes metadata" (Paragraph 26). Furthermore, the examiner wishes to refer to the Abstract and Column

6 of **Win** which state “the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource” (Abstract), “If the name and password are correct, the Authentication Client Module reads the user’s roles from the Registry server” (Column 6, lines 44-46), and “a personalized menu is an HTML page containing a list of authorized Resources” (Column 6, lines 13-14).

Moreover, the examiner further wishes to refer to Columns 12 and 17 of **Pazandak** which state “LL interface descriptor (B), as defined in more detail below, which is sent to the Parser Farm 308 on the server below” (Column 12, lines 19-22) and “The Parser 310 can also send the set or a subset of the Interface Descriptor 306, e.g., LL Parser ID, transaction ID, or other metadata” (Column 17, lines 16-20). The examiner wishes to state that the metadata in **Win** is akin to Security oriented user information (see “name and password” and “user roles”). The examiner further wishes to state that the metadata in **Pazandak** is akin to Security oriented user information as well (see ““The Parser 310 can also send the set or a subset of the Interface Descriptor 306, e.g., LL Parser ID, transaction ID, or other metadata”). The examiner further wishes to state that because both references teach metadata as defined by paragraph 26 of the specification of the instant application, both references subsequently teach “metadata”, and as a result, the use of both references is proper.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Mahesh Dwivedi



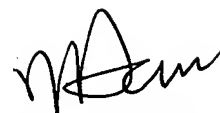
Patent Examiner

AU 2168

Houssain Alam

Supervisory Patent Examiner

AU 2166



HOSAIN ALAM
SUPERVISORY PATENT EXAMINER

Charles Rones

Supervisory Patent Examiner

AU 21634



CHARLES RONES
SUPERVISORY PATENT EXAMINER